

Available online at www.sciencedirect.com**SciVerse ScienceDirect**

Procedia Computer Science 17 (2013) 1047 – 1054

Procedia
Computer Science

Information Technology and Quantitative Management (ITQM2013)

An Embedded Backward Feature Selection Method for MCLP Classification Algorithm

Meihong Zhu^{*}, Jie Song*School of Statistics, Capital University of Economics and Business, Zhang Jia Lu Kou 121, Fengtai District, Beijing, 100070, China*

Abstract

Feature selection is very crucial for improving classification performance, especially in the case of high-dimensional data classification. Different classification algorithms tend to select different optimal feature subsets. Based on detailed analysis of the characteristics of Multiple Criteria Linear Programming (MCLP) classification algorithm, a feature selection criterion is presented and an embedded backward feature selection procedure is designed for MCLP in this paper. Experiments on four datasets (artificial and real-world) are carried out, and the effectiveness of the presented method is assessed. Results show that it achieves good performance as expected.

© 2013 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Selection and peer-review under responsibility of the organizers of the 2013 International Conference on Information Technology and Quantitative Management

Keywords: Multiple Criteria Linear Programming, Feature Selection, Embedded, Backward

1. Introduction

In the domain of classification, feature or variable selection (hereinafter referred to as FS) is very crucial for improving classification performance, especially in the case of high-dimensional data classification. By eliminating irrelative and redundant features, FS is conducive to improve learning speed, predictive accuracy and simplicity and comprehensibility of learned results [1, 2]. Although many feature selection methods have been presented, none of them can be suitable for all classification algorithms. Theoretical and experimental researches show that, different classification algorithms tend to select different optimal feature subsets. So, for a specific classification algorithm, we need to analyze its characteristics and find the corresponding FS methods

^{*}Corresponding author.

E-mail address: zhumh1027@126.com

for it. This paper intends to design an effective FS method for Multiple Criteria Linear Programming (MCLP) classification algorithm. It is organized as follows: Section 2 reviews some widely used FS methods. Section 3 describes the principle, model and properties of MCLP. In section 4, an embedded backward FS method for MCLP is proposed. In section 5, the effectiveness of this method is verified on four datasets. Finally, the availability of it is analyzed and concluded.

2. Feature selection

FS techniques are generally divided into three categories, depending on whether and how they combine the FS process with the learning algorithm: filter methods, wrapper methods and embedded methods [3-5].

When a FS method is used as a data preprocessing technique, it is often separated from the specific learning task or algorithm, and it is usually called a filter method. In a filter method, the goodness of a feature/feature subset is dependent on some general measure, such as distance measure, information measure, dependence measure, consistency measure or accuracy measures [4]. With this filter method, all features are ranked according to one or some of above measures, and only the highest ranking features are selected. The result of FS also relies on general characteristics of training data, but it does not involve any learning algorithm. A filter method also provides a very easy way to calculate and can simply scale to large-scale datasets since it only has a short computing time. However, the selected features may be not useful for a specific classification algorithm.

A wrapper method embeds a feature selection process within a classification algorithm. In a wrapper method, a search is conducted in the feature space, evaluating the goodness of each found feature subset by the estimation of the accuracy of the specific classifier to be used, training the classifier only with the found features. Because a wrapper method incorporates the interaction between feature selection and classification model, it is claimed that a wrapper method can get better predictive accuracy estimate than a filter method [2]. However, its computational cost is higher than a filter method and often it has a higher risk of over-fitting.

Compared with filter and wrapper, in an embedded method, the feature search process is embedded into the classification algorithm, and the learning process and the feature selection process can't be separated. Similar to a wrapper method, an embedded method includes the interaction with the classifier, while at the same time, it can save computation cost largely than a wrapper method.

3. MCLP classification algorithm

MCLP classification model was proposed by Shi [6, 7] based on optimization theory. A two-class MCLP classification model can be depicted as follows:

Here we have a training set T which includes n cases, r input features and a two-class target (output) feature labeled by "Bad" or "Good". Let $A_i = (A_{i1}, \dots, A_{ir})$ represent the i th case, where $i = 1, \dots, n$. MCLP aims at finding the best coefficient vector of r features $X = (x_1, \dots, x_r)^T$, and a boundary value b , to correctly classify each existing case into Bad or Good; that is,

$$A_i X \leq b, \forall A_i \in \text{Bad} \text{ and } A_i X \geq b, \forall A_i \in \text{Good}.$$

Due to incomplete linear separability of two classes in real-world datasets, some cases will be misclassified. Two kinds of distance are defined to measure the degree of misclassification or correct classification of the case A_i :

If A_i is misclassified, let α_i be the distance between A_i and boundary b ;

If A_i is correctly classified, let β_i be the distance between A_i and boundary b .

The objective of MCLP is minimizing $\sum_i \alpha_i$ and maximizing $\sum_i \beta_i$ simultaneously. The initial MCLP classification model M_1 is

$$\begin{aligned}
 & \text{Minimize } \sum_i \alpha_i \text{ and Maximize } \sum_i \beta_i \\
 (M_1) \quad & \text{Subject to :} \\
 & A_i X = b + \alpha_i - \beta_i, \forall A_i \in \text{Bad} \\
 & A_i X = b - \alpha_i + \beta_i, \forall A_i \in \text{Good}
 \end{aligned}$$

Where A_i is given; X and b are unrestricted; and $\alpha_i, \beta_i \geq 0$.

Since there exists conflict between the above two objectives, a compromise solution approach is introduced to balance them [6], and thus, this multi-objective linear programming model M_1 can be transformed into a single-objective model.

Let the “ideal value” of $-\sum_i \alpha_i$ be α^* (> 0) and the “ideal value” of $\sum_i \beta_i$ be β^* (> 0). Then four kinds of regret are defined to measure the difference between actual values and ideal values of the above two objectives. Detailedly, If $-\sum_i \alpha_i > \alpha^*$, the regret is defined as $-d_\alpha^+ = \alpha^* + \sum_i \alpha_i$; otherwise, $-d_\alpha^+ = 0$. If $-\sum_i \alpha_i < \alpha^*$, the regret is defined as $d_\alpha^- = \alpha^* - \sum_i \alpha_i$; otherwise, $d_\alpha^- = 0$. Thus, the following three formulae always hold: (i) $\alpha^* + \sum_i \alpha_i = d_\alpha^- - d_\alpha^+$, (ii) $|\alpha^* + \sum_i \alpha_i| = d_\alpha^- + d_\alpha^+$, and (iii) $d_\alpha^-, d_\alpha^+ \geq 0$. Similarly, the following three formulae also always hold: (i) $\beta^* - \sum_i \beta_i = d_\beta^- - d_\beta^+$, (ii) $|\beta^* - \sum_i \beta_i| = d_\beta^- + d_\beta^+$, and (iii) $d_\beta^-, d_\beta^+ \geq 0$. The above four regrets are hoped to be as little as possible. So M_1 can be changed into a single-objective linear programming model M_2 .

$$\begin{aligned}
 & \text{Minimize } Z = d_\alpha^- + d_\alpha^+ + d_\beta^- + d_\beta^+ \\
 & \text{Subject to :} \\
 (M_2) \quad & \alpha^* + \sum_i \alpha_i = d_\alpha^- - d_\alpha^+ \quad (1) \\
 & \beta^* - \sum_i \beta_i = d_\beta^- - d_\beta^+ \quad (2) \\
 & A_i X = b + \alpha_i - \beta_i, \forall A_i \in \text{Bad} \quad (3) \\
 & A_i X = b - \alpha_i + \beta_i, \forall A_i \in \text{Good} \quad (4)
 \end{aligned}$$

Where A_i, α^* , and β^* are given, X and b are unrestricted, and $\alpha_i, \beta_i, d_\alpha^-, d_\alpha^+, d_\beta^-, d_\beta^+ \geq 0$. In practice, α^* can be given as 0.00001, β^* as 99999, and b as 1.

In the MCLP classification model, the idea of linear programming is applied to determine a hyperplane for separating classes. The objective reflects its controlling on classification error, and classification discriminant criteria are included in the constraints. If X is solved, a labelled or unlabeled case A_i can be classified by the value of $A_i X$: A_i is classified as “Bad”, if $A_i X < b$; A_i is classified as “Good”, if $A_i X > b$.

Compared with other classification algorithms, MCLP is simple and direct, free of any assumptions, flexible in defining and modifying parameters, and high on classification accuracy. It has been widely used in business fields, such as credit card analysis, fraud detection, and so on.

4. An embedded backward feature selection method for MCLP

Supposing the best coefficient vector of all features $X = (x_1, \dots, x_r)^T$ has been obtained by implementing MCLP on the training set T . Applying the solution X on T , the corresponding optimal classification accuracy can be estimated. Now dimension reduction will be investigated in the premise of keeping the present optimal result.

From the model M_2 , It can be seen that, the effect mechanism of a feature F_j on classification result is that, it directly affects the constraints (3) and (4), and then affects the constraints (1) and (2), and finally the objective Z . For a case A_i , the effect of F_j on Z is $A_{ij}X_j$. In the constraints (3) and (4), the effects of F_j on them can be

demonstrated in the Table 1.

Table 1. The effect mechanism of F_j on the constraints (3) and (4)

Class	A_i is misclassified.		A_i is correctly classified.	
	F_j is not removed. ($X_j \in X$)	F_j is removed. ($X_j \notin X$)	F_j is not Removed. ($X_j \in X$)	F_j is removed. ($X_j \notin X$)
Bad	$A_i X = b + \alpha_i$	$A_i X - A_{ij} X_j = b + (\alpha_i - A_{ij} X_j)$	$A_i X = b - \beta_i$	$A_i X - A_{ij} X_j = b - (\beta_i + A_{ij} X_j)$
Good	$A_i X = b - \alpha_i$	$A_i X - A_{ij} X_j = b - (\alpha_i + A_{ij} X_j)$	$A_i X = b + \beta_i$	$A_i X - A_{ij} X_j = b + (\beta_i - A_{ij} X_j)$

Supposing the sample size of class “Bad” is n_1 , and that of class “Good” is n_2 . Based on the training result of the last time, m_1 cases are misclassified while $n_1 - m_1$ cases are correctly classified in class “Bad”; Similarly, m_2 cases are misclassified while $n_2 - m_2$ cases are correctly classified in class “Good”. The integrated effect of F_j on terms $\sum_i \alpha_i$ and $\sum_i \beta_i$ can be demonstrated in the Table 2 and the formulae (5) and (6).

Table 2. The integrated effect of F_j on $\sum_i \alpha_i$ and $\sum_i \beta_i$

Class	A_i is misclassified.		A_i is correctly classified.	
	F_j is not removed ($X_j \in X$)	F_j is removed. ($X_j \notin X$)	F_j is not removed ($X_j \in X$)	F_j is removed ($X_j \notin X$)
Bad	$\sum_{m_1} A_i X = \sum_{m_1} b + \sum_{m_1} \alpha_i$	$\sum_{m_1} A_i X - \sum_{m_1} A_{ij} X_j = \sum_{m_1} b + (\sum_{m_1} \alpha_i - X_j \sum_{m_1} A_{ij})$	$\sum_{n_1-m_1} A_i X = \sum_{n_1-m_1} b - \sum_{n_1-m_1} \beta_i$	$\sum_{n_1-m_1} A_i X - \sum_{n_1-m_1} A_{ij} X_j = \sum_{n_1-m_1} b - (\sum_{n_1-m_1} \beta_i + X_j \sum_{n_1-m_1} A_{ij})$
Good	$\sum_{m_2} A_i X = \sum_{m_2} b - \sum_{m_2} \alpha_i$	$\sum_{m_2} A_i X - \sum_{m_2} A_{ij} X_j = \sum_{m_2} b - (\sum_{m_2} \alpha_i + X_j \sum_{m_2} A_{ij})$	$\sum_{n_2-m_2} A_i X = \sum_{n_2-m_2} b + \sum_{n_2-m_2} \beta_i$	$\sum_{n_2-m_2} A_i X - \sum_{n_2-m_2} A_{ij} X_j = \sum_{n_2-m_2} b + (\sum_{n_2-m_2} \beta_i - X_j \sum_{n_2-m_2} A_{ij})$

$$\sum_{X_j \notin X} \alpha_i = (\sum_{m_1} \alpha_i - X_j \sum_{m_1} A_{ij}) + (\sum_{m_2} \alpha_i + X_j \sum_{m_2} A_{ij}) = (\sum_{m_1} \alpha_i + \sum_{m_2} \alpha_i) + (X_j \sum_{m_2} A_{ij} - X_j \sum_{m_1} A_{ij}) = \sum_i \alpha_i + X_j (\sum_{m_2} A_{ij} - \sum_{m_1} A_{ij}) \quad (5)$$

$$\sum_{X_j \in X} \beta_i = (\sum_{n_1-m_1} \beta_i + X_j \sum_{n_1-m_1} A_{ij}) + (\sum_{n_2-m_2} \beta_i - X_j \sum_{n_2-m_2} A_{ij}) = (\sum_{m_1} \beta_i + \sum_{m_2} \beta_i) + (X_j \sum_{n_1-m_1} A_{ij} - X_j \sum_{n_2-m_2} A_{ij}) = \sum_i \beta_i + X_j (\sum_{n_1-m_1} A_{ij} - \sum_{n_2-m_2} A_{ij}) \quad (6)$$

For the feature F_j , if it meets with formula (7) or condition (8), it also meets with formula (9).

$$X_j = 0 \quad (7)$$

$$\sum_{m_1} A_{ij} = \sum_{m_2} A_{ij} \text{ and } \sum_{n_1-m_1} A_{ij} = \sum_{n_2-m_2} A_{ij} \quad (8)$$

$$\sum_{X_j \notin X} \alpha_i = \sum_{X_j \notin X} \alpha_i \text{ and } \sum_{X_j \in X} \beta_i = \sum_{X_j \in X} \beta_i \quad (9)$$

This means that removing F_j does not change the constraints (1) and (2) and thus the objective Z . According to this idea, feature F_j meeting with (7) or (8) can be eliminated on the next round of training. In practice, there are few features which can exactly meet with (7) or (8). So only features approximately according with (7) or (8) can be found. It is hoped that formulae (10) and (11) are all as little as possible.

$$|(\sum_{m_1} A_{ij} - \sum_{m_2} A_{ij}) \times X_j| \quad (10)$$

$$|(\sum_{n_1-m_1} A_{ij} - \sum_{n_2-m_2} A_{ij}) \times X_j| \quad (11)$$

Here formula (10) indicates the effect of F_j on $\sum_i \alpha_i$, and formula (11) on $\sum_i \beta_i$. Considering the inconsistency between (10) and (11) in ranking features, they can be synthesized into one item. Because of the same importance between $\sum_i \alpha_i$ and $\sum_i \beta_i$ in objective Z , (10) and (11) should be the same important. Then, the feature elimination/selection criterion is designed as formula (12).

$$\lambda_j = |(\sum_{m_1} A_{ij} - \sum_{m_2} A_{ij}) + (\sum_{n_1-m_1} A_{ij} - \sum_{n_2-m_2} A_{ij})| \times |X_j| \quad (12)$$

According to the training results of the last round, λ values of all features are computed and ranked, and the feature with the smallest λ values can be removed in the next training round.

From the former item of formula (12), it can be seen that, λ_j reflects the separability of feature F_j on the two object classes. If the former item of formula (12) is large, it means F_j is an important feature for classification.

Compared with some other FS standards, such as Fisher ratio, λ_j reflects the effect of feature F_j on separating of the two object classes by considering misclassified cases and correctly classified cases respectively. X_j measures the sensitivity of the output with respect to F_j . If $|X_j|$ is small, then removing F_j does not produce large effect on the output. From the angle of feature selection process, this method is a backward FS method. According to feature search strategy, it applies a heuristic search strategy. Because the FS procedure is embedded into the classification process, it can be viewed as an embedded FS method.

5. Experimental arrangement and analysis

5.1. Data for analysis

Here four datasets are used to appraise the effect of the proposed FS method. The characteristics of them are described in the Table 3. In the dataset A, the first three features, F_1 , F_2 and F_3 , are purposely designed and all cases distributed in the space of the first three features as shown in the Figure 1. Another two features F_4 and F_5 are designed as $F_4=2F_1$ and $F_5=F_2+F_3$. Other 15 features are simply random variables whose values are between 0 and 1. According to general viewpoint, F_4 and F_5 are redundant features, and F_6 - F_{20} are all irrelative features. In the Sonar dataset, there are high correlations between two adjacent features and low correlations among other features. In the Credit Card and the Email User datasets, only low correlations exist among features.

Table 3. The characteristics of four datasets

Data set	Source	Size	Number of attributes	Size of class 1	Size of class 2
A	Artificial	200	20	100	100
Sonar	UCI	208	60	97	111
Credit card	Real-world	5000	66	815	4185
Email User	Real-world	9917	38	4354	5563

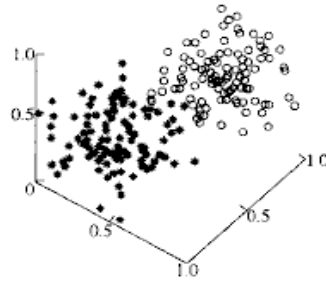


Fig. 1. The distribution of cases in the three-dimensional space of F_1, F_2 and F_3

5.2. Feature selection procedure for MCLP

Considering computing cost, the two large datasets--the Credit Card and the Email User, are randomly divided into training sets and test sets respectively. In the Credit Card dataset, 800(400+400) cases are used as training datum and the rest as test datum. In the Email User dataset, 2000(1000+1000) cases are used as training datum and the rest as test datum. FS procedures are implemented on the training sets and the final model evaluations are executed on the tests. Because of the randomness in partitioning of dataset into training set and test set, five rounds of partitioning are implemented on the Credit Card and the Email User respectively. The detailed operation is presented in Figure 2 by taking the example of the Credit Card dataset.

For $l=1$ to 5

Step1. Partition the Credit Card into training set TR_l and test set Te_l randomly.

$TR_{l1}=TR_l$ (The initial training set of l th partition is set as TR_{l1} .)

For $p=1$ to s_l (Loop of FS Process in l th partitioning)

Step2. Run MCLP on TR_{lp} , and solution X_{lp} is generated.

Step3. Compute $A_i X_{lp}$ for A_i in TR_{lp} , and decide whether A_i is correctly classified.

Step3. Put m_{1lp} misclassified "Bad" cases together and $n_{1lp}-m_{1lp}$ correctly classified "Bad" cases together; and Similarly, Put m_{2lp} misclassified "Good" cases together and $n_{2lp}-m_{2lp}$ correctly classified "Good" cases together.

Step4. Compute $\sum_{m_{1lp}} A_{ij}$, $\sum_{m_{2lp}} A_{ij}$, $\sum_{n_{1lp}-m_{1lp}} A_{ij}$ and $\sum_{n_{2lp}-m_{2lp}} A_{ij}$

for each feature F_j respectively.

Step5. Compute λ_{jlp} for each F_j respectively.

Step6. Rank all features according to their λ values.

Step7. Compute accuracy for class "Bad" and class "Good" and the whole accuracy on TR_{lp} . If they begin to degrade obviously, end the loop of p , go to step 10. Otherwise, go to step 8.

Step8. Remove the feature F^* with the smallest λ value.

Step 9. $TR_{lp}=TR_{lp}-F^*$

$p=p+1$

end of p .

Step10. Supposing the loop of p is ended in the s_l th iteration, then $k_l (=66-s_l)$ features are left in TR_l finally, and ranking weight for each of the k_l features is computed as formula (13):

$$w_{jl} = \frac{\lambda_{jls_l}}{\sum_{j=1}^{k_l} \lambda_{jls_l}}, \quad j = 1, 2, \dots, k_l \quad (13)$$

$l=l+1$
end of l

Fig. 2. FS procedure

After five loops of l are executed, a synthetical ordering measure for FS is defined as formula (14):

$$w_j = \frac{\sum_{l=1}^5 w_{jl}}{5}, \quad j = 1, 2, \dots, k (k \geq k_l) \quad (14)$$

The selected k_l features among five different partitions maybe are different, so k is not smaller than k_l . According to formula (14), k features are ranked and 30 ($< k$) features with larger W_j are selected in the end.

On the five original training sets- TR_1, TR_2, TR_3, TR_4 and TR_5 , the above 30 features are selected respectively, and five new classifiers C_1, C_2, C_3, C_4 and C_5 are trained. Applying them on the corresponding test sets TE_1, TE_2, TE_3, TE_4 and TE_5 , five partitions of test accuracy are computed and their average is used as the accuracy estimation of FS. It can be noted that, there exists very weak correlations among input features in the Credit Card dataset, so several features with small λ values can be simultaneously removed in step 8.

As for the Email User, only 15 features are selected finally from it.

For the two small datasets, A and Sonar, FS procedures are directly executed on them and the effects of feature selection are also measured on them. Because of the strong correlation among features, eliminating a feature will bring remarkable change of ranking of other features. So in each step of feature selection, only one feature with the smallest λ is eliminated.

5.3. Experimental results and analysis

According to experimental results on the four datasets, three aspects of important conclusions for MCLP are abstracted.

Firstly, MCLP has some similar characteristics to some other classification algorithms with respect to FS. When two input features are highly correlative, one of them is judged as redundant and is eliminated. In the artificial dataset A, F_4 is perfectly correlated with F_1 . So F_4 is eliminated at the first interaction of training. Similarly, F_2 is highly correlated with F_3 , and F_5 is the linear combination of F_2 and F_3 . Then one of the three features, F_2 , is redundant and is removed in the first interaction, too. Features with prominent difference (especially, mean) between two different classes are thought as important features for classification. If there exist correlations among features at some degree, removing one feature will bring change of ranking of other features. Under this circumstance, features only can be eliminated one by one while removing features in bulk at one step is unreasonable.

However, different to general conclusions, some random generated features with almost equal means between different classes are still selected into the final feature subset. For example, in the artificial dataset A, F_6 - F_{20} are all random generated features with almost equal means between two different classes, but some of them are still left in the training set in the end.

Finally, the FS criterion λ is very effective for MCLP. Executing FS with λ , the number of selected features in the end is significantly smaller than that of the original feature set. Modeling on the smaller feature subset, computational efficiency is improved while classification accuracy does not obviously degrade. The experimental results on the four datasets are shown in the Table 4.

Table 4. The experimental results on the four datasets

Dataset	A		Sonar		Credit Card		Email User	
Number of features	Before FS	After FS	Before FS	After FS	Before FS	After FS	Before FS	After FS
	20	8	60	30	66	30	38	15
accuracy of class 1	0.96	0.92	0.81	0.86	0.75	0.74	0.61	0.61
accuracy of class 2	0.90	0.92	0.87	0.82	0.79	0.78	0.69	0.67
Total accuracy	0.93	0.92	0.84	0.84	0.78	0.77	0.65	0.64

6. Conclusions

In this paper, an embedded backward FS method is specially designed for MCLP classification algorithm. Similar to some filter methods, the FS criterion λ_j reflects the importance of feature F_j on classifying the two object classes by considering misclassified cases and correctly classified cases respectively. At the same time, λ_j includes the coefficient X_j of F_j which represents the sensitivity of the output with respect to F_j . Experiments on one artificial dataset and three real-world datasets have been carried out. Results show that it achieves good performance and it is effective for MCLP. However, comparison of it with other popular feature selection criteria is not implemented and its advantage or disadvantage is not investigated in this paper.

Acknowledgement

This research has been financially supported by the Talent Improvement Plan of Beijing Education Committee --the Innovation Team Plan of Measurement and Research on Consumer's Confidence. It also has partially supported by a grant of National Natural Science Foundation of China (No. 11201316).

References

- [1] A. Blum and P. Langley. Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence*, 97:245-271, 1997.
- [2] R. Kohavi and G. John. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1-2):273-324, 1997.
- [3] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, vol. 3: 1157-1182, 2003.
- [4] G.Victo Sudha George and Dr. V.Cyril Raj. Review on Feature Selection Techniques and the Impact of SVM for Cancer Classification Using Gene Expression Profile. *International Journal of Computer Science & Engineering Survey (IJCSSES)* Vol.2, No.3, August 2011.
- [5] C.Lazar, J.Taminau J, S. Meganck and et.al.. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Trans Comput Biol Bioinform.* 2012 Jul-Aug; 9(4):1106-19.
- [6] Y. Shi and P. L. Yu. Goal Setting and Compromise Solutions. B. Karpak and S. Zionts (editors) , *Multiple Criteria Decision Making and Risk Analysis Using Microcomputers*. Berlin: Springer, 1989. 165–203.
- [7] Y.Shi, Y. Peng and et.al.. Data mining via Multiple Criteria Linear Programming: Applications in Credit Card Portfolio Management. *Information Technology and Decision Making*, 2002,1 (1) : 145-166.
- [8] UCI Irvine Machine Learning Repository, available online: <http://archive.ics.uci.edu/ml/>.